IN THE SPECIFICATION

Please amend the paragraph beginning on page 3, line 8, as follows:

In one embodiment, system 10 may be any processor-based system. Examples of the system 10 include a personal computer (PC), a hand held device, a cell phone, a personal digital assistant, and a wireless device. Those of ordinary skill in the art will appreciate that system 10 may also include other components, not shown in Figure 1 ~~3~~.

Please amend the paragraph beginning on page 5, line 25, as follows:

Figure 5 shows ~~a~~ hypothetical data in the register scoreboard 35 and the extended register scoreboard 40 for scheduling instructions 125 shown in Figure 4 according to one embodiment of the present invention. The dependency data 45 in the extended register scoreboard 40 and the register scoreboard 35 is shown in Figure 5 for the code piece in Figure 4. The extended register scoreboard 40 and the register scoreboard 35 use data-dependency-stall number (DDSN) $I_{m,n}$ (where m is the m-th instruction and n is the n-th one) instead of true-or-false boolean value for every two instructions. In one embodiment, the DDSNs are the maximum possible pipeline stall cycles between two instructions. In the extended register scoreboard 40 and the register scoreboard 35, a negative number "-1" stands for no data dependency between two instructions.

Please amend the paragraph beginning on page 6, line 23, as follows:

In Figure 6, for every non-zero GAP, the first loop (code lines 2-9 ~~2~~-~~9~~) searches the previous instructions before G_C of this GAP, until the GAP has been fully filled. If the current instruction is encapsulated by another GAP (code line 3), or it has been moved before (code line 4), the loop will break. If DWN of the current instruction is larger than G_C, the current instruction will be moved before the next instruction after G_C (code line 6). The L0 of the moved instruction will be subtracted from GAP (code line 7).

Please amend the paragraph beginning on page 6, line 29, as follows:

The second loop (code lines 11-18 ~~11~~-~~18~~) searches the instructions behind the current GAP. The loop and break conditions (code lines 11, 12, 13) are similar to the aforementioned loop. The UP instead of DWN is used in the condition at code line 14.

And the movable instructions are moved after the instruction before GAP (code line 15). All instructions in a code block are searched at most twice and there is no need to update any information except non-zero GAPs. Hence, the complexity of this heuristic rule is linear.